REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested.

Upon entry of this reply, claims 1-32 will remain in the application.

Claims 1-6, 11-13, 15-20 and 32 were rejected as being allegedly anticipated under 35 U.S.C. § 102 by U.S. Patent No. 6,405,289 to Arimilli, et al. ("Arimilli patent" or "Arimilli"). Claims 7-10, 14, and 21-31 were rejected as being allegedly obvious under 35 U.S.C. § 103 in light of Arimilli.

Applicants teach a technique for extending the local memory address space of a processor. In an embodiment, a processor may include a local addressable memory, such as an SRAM, in parallel with L1 local cache. A local memory controller may examine a local memory descriptor to determine whether a page containing a requested memory location is in the local addressable memory. If the requested memory location is not in the local addressable memory, the local memory controller may route the access to the local cache instead.

Arimilli discusses a coherent memory system in a symmetric multiprocessor computer. The processors communicate with each other before each read or write to a memory location. As part of this communication, a processor polls its own cache, e.g., L1 cache.

The Office Action illustrates two mistaken assumptions about the nature of the claimed "local addressable memory." First, the Office Action states that the claimed "local addressable memory" is the same as ordinary cache memory. For example, on page 8, the Office Action states that "one of

[Aramilli's] local level caches (L1 or L2) can be considered to be equivalent to the claimed local addressable memory."

Second, the Office Action equates "accessing a local addressable memory in response to the local memory descriptor indicating that the memory block is in the local addressable memory," as recited in claim 1, with polling the cache as in Arimilli. For example, the Office Action states on page 8, "the concept of accessing the local addressable memory . . . is equivalent to the polling operation which yields [a] cache hit."

Applicants respectfully point out the true scope of local addressable memory below. This discussion focuses on the specification to demonstrate that no new matter is being added.

The claimed "local addressable memory" is not ordinary cache.

The Office Action equates the claimed "local addressable memory" of the present application with ordinary cache, which is illustrated in Arimilli. "Local addressable memory," however, is different in at least three fundamental ways: 1) cache is not directly addressed; 2) cache is not programmable; and 3) accessing cache entails both cache hits and cache misses.

First, cache is not directly accessed as a separate physical memory address. Instead, cache intercepts accesses to a memory address; the cache itself is not located at a particular memory address. Moreover, the cache does not contain the requested data, but only a copy of the data. The specification illustrates these concepts in paragraph 0011: "[a] memory controller 135 may check the address of the requested memory location and send the access to the L1 cache 115. If the L1 cache 115 has a copy of the requested

information (cache hit), the L1 cache returns the requested information."

A memory access, therefore, is not directed at cache memory, because cache is not the physical location that the memory request wishes to access. Rather, the request is sent to cache because the cache might have a copy of the data from the desired physical memory location. Thus, the cache acts as a temporary mirror of a physical memory location.

In contrast, the claimed "local addressable memory," e.g., L1 SRAM described in the specification, is directly addressed and is different from cache. The application makes this clear in paragraph 0012: "[u]nlike the L1 caches, the L1 SRAMs are 'real' memory." The specification thus makes clear in unambiguous language that L1 SRAM is different from L1 cache. Unlike the cache, L1 SRAM is directly addressed, as described in paragraph 0013: "Some of the system memory may be mapped in the L1 memory address space." Local addressable memory is thus directly mapped to particular addresses, whereas cache is not accessed outside of its role as a temporary copy of data at a particular address.

Second, because cache is not directly accessed, and because it has no role in the memory scheme divorced from physical memory, cache is not programmable with instructions and data. In contrast, paragraph 0012 of the specification notes that "[t]he L1 SRAM 145 may be programmed with instructions and data used in, for example, DSP-critical applications." Cache, cannot be said to be "programmed with instructions and data" because it only holds temporary copies of data that exists and is addressed at other locations.

Third, accessing the cache entails both cache hits and cache misses. See, e.g., paragraph 0011 of the specification.

However, "accesses to L1 SRAM may not entail cache misses and the associated penalties." Application at ¶ 0012.

Cache misses are inherent in any scheme of accessing data in a cache. The fact that local addressable memory does not suffer from cache misses is yet another distinction between cache memory and local addressable memory. For at least these reasons, the claims are allowable over Arimilli.

Accessing local addressable memory is different than polling a cache.

The Office Action also incorrectly equates accessing local addressable memory with polling a cache. As described in David A. Patterson & John L. Hennessy, Computer Architecture A Quantitative Approach 377-378 (Morgan Kaufmann Publishers 1996) (1990), polling a cache involves checking the cache for whether it contains a copy of the data that is at a desired physical address. This is done by checking the address tags that are associated with each block frame of the cache. These tags indicate which physical addresses the cache is currently mirroring; the cache temporarily holds a copy of the data that is at these addresses.

Thus, when a memory access is performed, the cache is polled by checking the address tags. A cache hit occurs when the address tags indicate that the cache contains a copy of the data from the desired physical address. Conversely, a cache miss occurs when the address tags indicate that the cache does not contain a copy of the data from the desired physical address.

In the case of a cache miss, the main system memory responds to the memory access. The cache copies the data from

the main memory, and as it does so, it writes over data mirroring other physical addresses.

This procedure is different from accessing local addressable memory as outlined in the specification. As described in paragraphs 0013-0017, a local memory descriptor identifies whether a particular memory location is mapped to the local addressable memory. If the local memory descriptor indicates that a desired memory location is mapped to the local addressable memory, then all memory accesses to that address are directed to the local addressable memory. Thus, the local addressable memory does not simply hold a copy of the data from a particular address, as does a cache; rather, the local addressable memory holds original data itself.

If the local memory descriptor indicates that the desired memory location is not mapped to the local addressable memory, memory accesses may then be processed in a normal fashion, i.e., to a cache and, if necessary, to main memory. However, this process does not result in any writing over of data in the local addressable memory, as occurs in a cache.

This process is different from the traditional cache hit/miss scheme. In a cache, address tags are checked in order to determine whether the cache has a copy of the data from a particular address location; the actual data resides elsewhere in memory. In the local addressable memory, local memory descriptors are checked to determine whether the local addressable memory is the memory address being referenced. The actual data resides in the local addressable memory itself.

Moreover, a cache miss results in a copy of data from the main memory being written into the cache, overwriting other data. In contrast, an indication that a memory address in not mapped to the local addressable memory results in no overwriting

of data in the local addressable memory; rather the memory access is merely passed on to other portions of the memory architecture (e.g., cache, main memory, etc.).

In addition, the specification makes clear that although a non-existent memory access might be sent to the local addressable memory, this are not done in the proper course of events. For example, the specification states that "[i]f the access is to non-existent L1 SRAM memory, the local memory controller 135 may trigger an illegal-access violation exception." This is far different from a cache miss, which is not an error, but simply an expected part of working with a cache.

For at least these reasons, accessing the local addressable memory is very different from polling a cache, and the claims are therefore allowable over Arimilli.

Conclusion

As illustrated above, the local addressable memory of the present application is different from the cache of Arimilli and from any ordinary cache memory. Because the Office Action predicates all the claim rejections on this concept, Applicants respectfully request a reconsideration of the final rejections of claims 1-32.

Attorney's Docket No.: Intel 10559-566001
/ P12728

Applicant asks that all claims be allowed. No fees are believed to be due at this time. Please apply any additional charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: April 23, 2004

Scott C. Harris
Reg. No. 32,030

Attorneys for Intel Corporation

/BY
KENYON JENCKES
REG. NO. 41,873

Fish & Richardson P.C.
PTO Customer Number: 20985
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099
10388743.doc